

11 means for providing the clear portion to memory
12 locations accessible by a processor; and
13 remainder memory for storing the remainder portion
14 of the secure program, the remainder memory not directly
15 accessible by the processor;
16 means for requesting subsets of the remainder portion
17 for use by the processor; and
18 means, within the security chip, for checking that the
19 requested subset is a ^{predetermined} subset ~~expected to be requested given a~~ ^{dependent on the}
20 stored state for the processor.

1 2. (Once Amended Herein) The apparatus of claim 1,
2 wherein the secure program stored in the program memory is stored
3 with the [program] clear portion and the remainder portion stored
4 separately.

1 3. (Unamended) The apparatus of claim 1, wherein the
2 remainder portion is a set of branch instructions of the secure
3 program.

1 4. (Unamended) The apparatus of claim 3, wherein the
2 security chip further includes means for caching branch
3 statements based on recently executed branches.

1 5. (Once Amended Herein) The apparatus of claim 1,
2 wherein the means for decrypting [branches] portions of the
3 secure program is configured with a decryption key.

1 6. (Once Amended Herein) The apparatus of claim 5,
2 wherein the decryption key is stored in a volatile memory.

1 7. (Unamended) The apparatus of claim 6, wherein the
2 volatile memory is distributed over the security chip, the

3 security chip further comprising overlying circuitry which
4 overlies and obscures at least a part of the volatile memory.

1 8. (Once Amended Herein) The apparatus of claim 7,
2 wherein the overlying circuitry is coupled to a power source for
3 the volatile memory such that the removal of the [overly]
4 overlying circuitry removes the power to the overlying circuitry.

1 9. (Once Amended Herein) The apparatus of claim 1,
2 further comprising:

3 [a] clocking means, within the security chip, for
4 determining a rate of instruction execution of the processor[,];
5 and

6 [wherein the security chip responds to] timing response
7 means for rejecting processor requests [only] when the clocking
8 means determines that the rate is [within an expected] outside a
9 range of normal operation for the processor.

no? 1 10. (Unamended) The apparatus of claim 1, further
2 comprising a data decompressor for decompressing the secure
3 program after decryption, wherein the secure program is
4 compressed before encryption.

no? 1 11. (Unamended) The apparatus of claim 10, wherein the
2 decompressor is an entropy decoder.

AH 1 12. (Once Amended Herein) The apparatus of claim 1,
2 further comprising:
3 [a] checksum means, within the security chip, for
4 determining a checksum of bus accesses on a processor bus[,]; and
5 [wherein the security chip responds to] checksum
6 response means for rejecting processor requests [only] when the
7 [determined] checksum [matches an expected checksum] does not
8 match a predetermined checksum for those bus accesses.

no? 1 13. (Unamended) The apparatus of claim 1, further
2 comprising a data scrambler for reordering data elements of the
3 secure program according to a reversible and deterministic
4 pattern determined by a key value, wherein the secure program is
5 reordered by the inverse of the data scrambler before being
6 placed in the program memory.

no? 1 14. (Unamended) The apparatus of claim 13, wherein the
2 data scrambler comprises a plurality of first-in, first-out
3 buffers.

no? 1 15. (Unamended) The apparatus of claim 13, wherein the
2 reversible and deterministic pattern is generated by reference to
3 the output of a pseudorandom number generator.

1 16. (Once Amended Herein) The apparatus of claim 1,
2 wherein the means for decrypting portions of the secure program
3 operates based on the key value and the output of a pseudorandom
4 number generator.

15 1 17. (Once Amended Herein) The apparatus of claim 1,
2 further comprising means for altering the operation of the
3 security chip and the program flow of the secure program when
4 said means for checking detects that an [unexpected] ~~invalidly~~ ^{the}
5 requested subset is not within the valid predetermined set of subsets
6 has been requested, [where by] whereby the
7 altered operation causes a negative effect on the program flow or
operation.

no? 1 18. (Unamended) The apparatus of claim 17, wherein the
2 means for altering is a means for halting the processor.

1 ^{to prevent} 19. (Unamended) An apparatus for ^{encrypting} ~~securing~~ program data
2 ~~from~~ unauthorized copying, comprising;

37

3 a branch separator for extracting branch statements
4 from the program data;
5 a compressor for compressing the extracted branch
6 statements and a remainder of the program data to form compressed
7 data; and
8 an encryptor for encrypting the compressed data.

1 20. (Once Amended Herein) [The apparatus of claim 19,
2 wherein the branch separator comprises:] An apparatus for
3 ^{encrypting} ~~securing~~ ^{to prevent} ~~program data from~~ unauthorized copying, comprising;
4 a branch separator for extracting branch statements
5 from the program data comprising:
6 means for automatically generating checksum data
7 representing checksums of program data; and
8 means for automatically generating timing information
9 used to assess timing of program data processing[,];
10 a compressor for compressing the extracted branch
11 statements, a remainder of the program data, [whereby] the
12 checksum data, and the timing information, [are compressed by the
13 compressor and encrypted by the encryptor] to form compressed
14 data; and
15 an encryptor for encrypting the compressed data.

1 21. (Once Amended Herein) A method of executing a
2 secure program to prevent copying of the secure program in a
3 usable form from information acquired over an insecure processor
4 bus, the usable form being a copy which replaces the
5 functionality of the original, comprising the steps of:
6 accepting a request from [the insecure] a processor
7 over the insecure processor bus for a block of program data, the
8 block of program data including at least one of one or more
9 program instructions or one or more program data elements;
10 decrypting, in a secure manner, the block of program
11 data into a clear portion and a remainder portion;

12 providing the clear portion to the [insecure] processor
13 over the insecure processor bus; and

14 accepting requests from the [insecure] processor over
15 the insecure processor bus for elements of the remainder portion;

16 checking that the request is [proper given] consistent
17 with the state of the [insecure] processor and previous requests;

18 processing the requests from the [insecure] processor
19 for elements of the remainder portion; and

20 responding to the requests with request responses,
21 wherein the request responses do not contain enough[, wherein
22 underlying remainder portion elements are not feasibly determined
23 by reference to only the information content [of a response to a
24 request] to recreate the remainder portion with ^{substantially} ~~less~~
25 computational effort than required to create ^{said remainder portion} ~~the secure program~~.

Conceded
1 22. (Once Amended Herein) The method of claim 21,
2 further comprising the steps of:

3 separating a program into the clear portion and the
4 remainder portion to form a secure program; and

5 encrypting the secure program prior to placing the
6 secure program [in a memory accessible by attackers intent on
7 making unauthorized copies of the secure program] into an
8 insecure memory.

1 23. (Once Amended Herein) The method of claim 22,
2 wherein the step of separating is a step of separating branch
3 instructions of the [secure] program from other instructions of
4 the [secure] program.

me 1
2 24. (Unamended) The method of claim 21, wherein the
step of decrypting is performed with a decryption key.

3d

1 25. (Once Amended Herein) The method of claim 24,
2 further comprising the step of storing the decryption key in a
3 volatile memory.

1 26. (Once Amended Herein) The method of claim 25,
2 further comprising the steps of:
3 providing a power source to the volatile memory;
4 covering the volatile memory with a circuit such that
5 the power source is removed from the volatile memory when the
6 circuit is disturbed and **[the contents of the volatile memory**
7 **cannot be easily measured without removing]** the circuit shields
8 the volatile memory from probing.

1 27. (Once Amended Herein) The method of claim 21,
2 further comprising the step of checking a rate of instruction
3 execution of the processor prior to providing a request response
4 **[to a request for information].**

1 28. (Unamended) The method of claim 21, further
2 comprising the step of decompressing the secure program after
3 decryption, wherein the secure program is compressed before
4 encryption.

1 29. (Once Amended Herein) The method of claim 21,
2 further comprising the steps of:
3 determining a checksum of bus accesses on a processor
4 bus;
5 comparing the checksum to a precalculated checksum
6 **[expected]** for a set of [the] instructions of the secure program
7 which **[were expected to be executed]** are executed under normal
8 operation; and
9 preventing the unobstructed operation of the secure
10 program when the checksum and the precalculated checksum differ.

1 30. (Once Amended Herein) The method of claim 21,
2 further comprising [a] the steps of:

3 scrambling an order of data elements of the secure
4 program according to a reversible and deterministic pattern
5 determined by a key value prior to storage in [a **memory**
6 **accessible by attackers**] an insecure memory; and

7 descrambling the order of the data elements upon proper
8 request of the processor.

1 31. (Unamended) The method of claim 30, wherein the
2 step of scrambling comprises a step of generating a pseudorandom
3 number used to form the reversible and deterministic pattern.

1 32. (Once Amended Herein) A method for ^{encrypting} ~~securing~~ a
2 program ~~against~~ ^{to prevent} unauthorized copying, comprising the steps of:
3 separating program code according to sequences of
4 nonbranch instructions and branch instructions;
5 compressing the [non-branch] nonbranch instructions to
6 form a first set of compressed data;
7 compressing the branch instructions to form a second
8 set of compressed data; and
9 encrypting the first and second sets of compressed
10 data.

Please enter new claim 33 as follows:

1 --33. (NEW) An apparatus for executing a secure
2 program in an insecure computer system, wherein the ability to
3 make workable copies of the secure program during execution of
4 the secure program using the insecure computer system is
5 inhibited, a workable copy being a copy which replaces the
6 functionality of the original secure program, the apparatus
7 comprising: